Tecnologías de la Traducción: Limitaciones del *modelo cero*[*]

Felipe Sánchez Martínez, Mikel L. Forcada,
Carlos Pérez Sancho, Antonio Pertusa Ibáñez

Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant

Introducción

Una aproximación preliminar —y bastante rudimentaria— a la traducción automática es la llamada *traducción palabra por palabra*: el sistema lee el texto original palabra a palabra de izquierda a derecha, sustituye cada palabra original por un equivalente fijo (de una palabra, de más palabras, o incluso de cero palabras) en lengua meta sin tener en cuenta el contexto y escribe las palabras una a una y en el mismo orden en el texto meta (esta aproximación la denominaremos en clase "modelo 0"[1]). Por ejemplo, si la frase tiene N palabras,

$$m_1\ m_2\ m_3\ \cdots\ m_N$$

la traducción *modelo cero* es

$$T(m_1)\ T(m_2)\ T(m_3)\ \cdots\ T(m_N)$$

donde $T(m)$ representa el equivalente fijo de la palabra m en la lengua meta, que puede tener cero, una o más palabras. Por ejemplo, la traducción *modelo cero* al inglés de la frase en español *Este ejercicio práctico es muy sencillo*, con $m_1 =$ *Este*, $m_2 =$ *ejercicio*, etc., podría ser *This exercise practical it is very simple* (incorrecta), donde, por ejemplo, $T(m_4) = T($*es*$) =$ *it is*.

[1] ¡Este nombre no lo encontraréis fuera de la asignatura!

Actividad

En esta actividad se deben identificar algunos de los problemas de esta aproximación y pensar en posibles estrategias de mejora. Para ello tendréis que forzar al sistema que habéis elegido para que traduzca palabra por palabra. Por ejemplo, escribiendo cada palabra de la frase en un párrafo independiente (con una línea en blanco completa entre palabra y palabra), en algunos casos con mayúscula inicial y acabado en punto, como si formara una frase por sí misma. Algunas de las frases que podéis probar a traducir y su traducción son:

1. (en) *Tony books a room in the hotel* → (es) **Tony libros un habitación dentro el hotel*
2. (en) The expert's large table is full → (es) **El experto - grande mesa es lleno*
3. (en) *The red houses are more expensive* → (es) **El rojo casas es más caro*
4. (en) *As a matter of fact, he picked it up* → (es) **Cuando un asunto de hecho, él elegido él arriba*

Pista: Para descubrir como mejorar las traducciones del *modelo cero* pensad en qué información se tendría que asociar a cada palabra para poder generar traducciones más adecuadas. ¿Cómo se usaría esta información?

Notas para seguir la explicación del profesor

Las traducciones *modelo cero* de los ejemplos se han obtenido con el traductor inglés–español de Apertium.org. Cuando se traducen las frases completas, el resultado es una traducción razonable.

Descripción de los problemas y esbozo de la solución:

1. En la primera oración hay dos palabras homógrafas, *books*, que puede tener dos traducciones: *reserva*, cuando es verbo, o *libros*, cuando es sustantivo e *in*, que se puede traducir por *en*, cuando es preposición, o por *dentro*, cuando es adverbio. Cuando el sistema ha traducido *books* aisladamente, lo ha interpretado como sustantivo, y cuando ha traducido *in* aisladamente, lo ha interpretado como adverbio.

   Para poder decidir en qué caso nos encontramos, el sistema tendría que usar información que no está presente en el texto. Por ejemplo, tendría que decidir que *books* es un *verbo* porque viene precedido de *Tony* que es un *nombre propio*. Para poder obtener la información (de que *books* puede ser sustantivo

o verbo, o de que *He* es un pronombre personal sujeto), el sistema necesita realizar *análisis morfológico*, y asignar a cada palabra una o más *categorías léxicas* o *partes de la oración*, y después usar reglas para decidir en los casos ambiguos (la misma regla se podría aplicar a *cooks* en *He cooks*). En el caso de *in*, el contexto, es decir, el sustantivo *room* que lo precede y el artículo *the* que lo sigue, permite elegir como más adecuada la interpretación según la cual *in* es una preposición.

Claramente, el sistema ya no puede trabajar directamente con el texto fuente tal como le llega. Para resolver los problemas asociados a la *homografía con cambio de categoría* en un sistema de TA basado en reglas o en conocimiento lingüístico, es necesario trabajar con una representación *abstracta* que resulta de un cierto nivel de *análisis* lingüístico, y que permite aplicar reglas *generales* para resolver los problemas. La traducción deja de ser directa y pasa a ser *indirecta*.

Si usamos la analogía de que traducir es como atravesar un río, la traducción directa tiene el problema que hay que atravesarlo nadando, mojándose. El análisis es como la escalera que sube a un puente que pasa por encima del agua. La abstracción es la elevación que nos permite cruzar el río fácilmente sin tener que bañarnos.

Cuando la ambigüedad léxica es de otra naturaleza, por ejemplo cuando la homografía no está asociada al cambio de categoría léxica,

(es) *vendo* $\rightarrow$ *vender*.verb.indic.pres.p1.sg

(es) *vendo* $\rightarrow$ *vendar*.verb.indic.pres.p1.sg

o cuando se trata de palabras *polisémicas* (la *polisemia* una propiedad de todas las formas flexionadas de un *lema* determinado) que pueden tener más de una traducción:

(ca) *estació* $\rightarrow$ (en) *station*

(ca) *estació* $\rightarrow$ (en) *season*

(ca) *estació* $\rightarrow$ (en) ...

la solución es más difícil.

2. En la segunda oración, queda claro que el orden de las palabras que produce el *modelo cero* es inadecuado porque para obtener una traducción adecuada hay que cambiar el orden de las palabras. ¿Cómo sabemos que las palabras *large table* se tienen que traducir por *mesa grande* y no por *grande mesa*? De nuevo, saber que *large* es un adjetivo y *table* es un sustantivo (en este contexto, puesto que puede ser una forma del verbo *to table*), nos permitiría escribir una regla del estilo de:

   (en) **adj subst** → (es) **subst adj**

   que nos permitiría tratar otros reordenamientos como por ejemplo *red car* → *coche rojo* o *institutional message* → *mensaje institucional*.

   Para poder aplicar estas reglas de reordenamiento, de nuevo es necesario el *análisis* (morfológico en este caso, seguido de desambiguación de palabras como *table*), es decir, la *abstracción* que nos permite tratar muchos problemas similares de reordenamiento con la misma regla.

   De hecho, para reordenar adecuadamente el sintagma nominal *the expert's large table* haría falta una regla más larga:

   (en) **det**$_1$ **subst**$_1$ **gs adj subst**$_2$ →
   (es) "el" **subst**$_2$ **adj** "de" **det**$_1$ **subst**$_1$

   donde los subíndices se usan para indicar mejor la operación de reordenamiento, "**gs**" representa el sufijo genitivo sajón (*'s*, *s'* , *'*) y las palabras entre comillas representan palabras que se tienen que añadir para construir la traducción.

3. En la tercera oración, se observa que el modelo *cero* no puede garantizar la concordancia, en este caso de género y número, de los determinantes *El* (→*las*) y los adjetivos *rojo* (→ *rojas*) y *caro* (→ *caras*) con los sustantivos. La misma noción de concordancia sólo se puede expresar en términos de conceptos *abstractos* como por ejemplo *determinante*, *adjetivo*, *sustantivos*, por lo tanto, aquí también queda clara la necesidad de *análisis*.

   Por ejemplo, las reglas para establecer la concordancia en el sujeto de la oración las podríamos formular sobre la regla de reordenamiento correspondiente:

   (en) **det adj subst** → (es) **det subst adj**
   asigna género meta: **subst** → **det**
   asigna género meta: **subst** → **adj**
   asigna número meta: **subst** → **det**
   asigna número meta: **subst** → **adj**

Con reglas de este tipo se puede tratar bien la concordancia *a la corta*, o *de contacto*.

Para asegurarnos de que el segundo adjetivo de la oración también concuerda tendríamos que escribir una regla más larga que lo incluyera, o inventarnos alguna manera de propagar el género y el número hacia adelante.

Un buen catálogo de patrones de reordenamiento y concordancia puede servir para tratar los casos más frecuentes y conseguir una calidad razonable de traducción.[2]

4. En este ejemplo hay dos *unidades léxicas de más de una palabra* (ULMP), las cuales, a pesar de estar compuestas por más de una palabra, se deben traducir como si fueran una unidad:

   (en) *as a matter of fact* $\rightarrow$ (es) *de hecho*
   (en) **pick** X *up* $\rightarrow$ (es) **coger** X

   Dado que el *modelo cero* traduce palabra a palabra, el resultado es inadecuado.

   La solución, en el primer caso, es relativamente sencilla: como la ULMP es invariable, lo que tiene que hacer el programa es dividir el texto en unidades léxicas de manera más *inteligente*, teniendo en cuenta un diccionario de ULMP, y tratar la ULMP como si fuera una sola palabra.

   En el segundo caso, hay dos problemas adicionales: por un lado, la ULMP no es invariable (**pick** puede aparecer de varias formas: *pick*, *picks*, *picked*, *picking*); por otro lado, la ULMP no es *contigua*: puede contener material en medio. Sin entrar en detalles, queda claro que la solución de los dos problemas sólo es posible con un cierto nivel de análisis (cómo se ve, la misma regla de arriba ya está formulada en términos más abstractos).

   Ejemplos entre el castellano y el catalán:

   (es) *con cargo a* $\rightarrow$ (ca) *a càrrec de*
   (ca) **trobar** *a faltar* $\rightarrow$ (es) **echar** *de menos*

Estudiaremos ahora en detalle la solución que se ha esbozado anteriormente.

[2] Los programas que suelen aplicar estos patrones lo hacen usando la estrategia LRLM ("left-to-right longest match"): es decir, van de izquierda a derecha, y aplican el patrón más largo que concuerda con la entrada; si no pueden aplicar una regla, traducen una palabra suelta (*modelo cero*!) y lo vuelven a intentar.

Simplificación de los diccionarios: Un efecto secundario interesante del uso del análisis es que los diccionarios bilingües, que en el *modelo cero* tendrían que incluir todas las formas flexionadas de cada palabra, se simplifican. Por ejemplo, en un diccionario español–catalán de *modelo cero* deberíamos introducir todas las formas del verbo maldecir:

(es) maldigo → (ca) maleïsc
(es) maldices → (ca) maleïxes
...
(es) maldecíamos → (ca) maleíem
...

mentres que si trabajamos con las formas analizadas quizás sólo necesitaríamos:

(es) *maldecir*.verbo → (ca) *maleir*.verbo

y de la información de flexión de la lengua origen se encargaría el proceso de análisis (de la lengua meta lo veremos a continuación).

El "modelo 1" y la traducción automática indirecta por transferencia: La solución descrita anteriormente (más avanzada que el modelo palabra por palabra pero todavía con numerosos problemas), la denominaremos *Modelo 1*. No es demasiado diferente de la que usan algunos programas de traducción automática como por ejemplo *Power Translator* (hasta la versión 5), *SDL Transcend*, *Reverso*, *interNOSTRUM* o *Apertium*. Se corresponde con un caso particular del que en la bibliografía se denomina *sistema de traducción automática indirecta por transferencia*, con tres módulos diferenciados:

Análisis: módulo monolingüe que hace el análisis morfológico, la desambiguación léxica categorial e identifica los patrones (secuencias de palabras) que hay que tratar. Genera una *representación abstracta del texto origen* (RATO).

Transferencia: módulo bilingüe que lee la RATO, gestiona el diccionario bilingüe y aplica las acciones de reordenamiento y concordancia asociadas a cada patrón identificado en la fase de *análisis*. Genera a partir de la RATO una *representación abstracta del texto meta* (RATM).

Generación: módulo monolingüe que genera, a partir de los lemas y las características morfológicas (flexión), las formas superficiales necesarias. Genera, a partir de la RATM un *texto concreto*: la traducción en bruto.

Esquemáticamente,

$$\text{TO} \rightarrow \boxed{\text{A}} \rightarrow \text{RATO} \rightarrow \boxed{\text{T}} \rightarrow \text{RATM} \rightarrow \boxed{\text{G}} \rightarrow \text{TM}$$

Es decir, el *análisis* identifica y agrupa situaciones que requieren un tratamiento diferenciado de traducción; la *transferencia* las transforma a una forma equivalente pero de la lengua meta; finalmente, la *generación* genera texto meta.

El *modelo 1* propuesto como respuesta a los problemas anteriores es, por tanto, un sistema de transferencia sintáctica parcial (no hace el análisis sintáctico completo en forma de árboles, por ejemplo, pero sí que identifica patrones). Hay sistemas de transferencia donde el análisis es más profundo y puede ser incluso semántico.

Modularidad: La *modularidad* de los sistemas de traducción automática indirecta por transferencia (de todos, no sólo del *modelo 1*) hace que podamos cambiar, por ejemplo, de lengua meta cambiando sólo los módulos de transferencia y de generación, o de lengua origen cambiando sólo los módulos de análisis y transferencia. Por ejemplo, si tenemos un traductor del inglés al español:

$$\text{en} \rightarrow \boxed{\text{A}_{\text{en}}} \rightarrow \boxed{\text{T}_{\text{en}\rightarrow\text{es}}} \rightarrow \boxed{\text{G}_{\text{es}}} \rightarrow \text{es}$$

podemos aprovechar el módulo de análisis del inglés A_{en} para hacer un traductor del inglés al catalán:

$$\text{en} \rightarrow \boxed{\text{A}_{\text{en}}} \rightarrow \boxed{\text{T}_{\text{en}\rightarrow\text{ca}}} \rightarrow \boxed{\text{G}_{\text{ca}}} \rightarrow \text{ca}$$

o el módulo de generación del español G_{es} para hacer un traductor del neerlandés al español

$$\text{nl} \rightarrow \boxed{\text{A}_{\text{nl}}} \rightarrow \boxed{\text{T}_{\text{nl}\rightarrow\text{es}}} \rightarrow \boxed{\text{G}_{\text{es}}} \rightarrow \text{es}$$

Cadena de montaje: El sistema de transferencia funciona como una *cadena de montaje*: dado que los tres módulos trabajan de izquierda a derecha y de una única pasada, no hace falta que un módulo espere a que el anterior acabe con el texto: pueden trabajar paralelamente; esto hace que los sistemas de esta naturaleza sean muy rápidos.

Reversibilidad parcial: Si se separan los datos lingüísticos usados por cada uno de los tres módulos, es posible una *reversibilidad parcial*. Si en el sistema inglés–español anterior separamos los *datos lingüísticos* de cada uno de los tres módulos del *software* que procesa estos datos (dA_{en}, $\text{dT}_{\text{en}\rightarrow\text{es}}$, dg_{es}) podemos definir un *motor* genérico de traducción (A, T, G) que sirve para cualquier par de lenguas:

$$\text{en} \rightarrow \boxed{\begin{matrix}\text{dA}_{\text{en}}\\ \text{A}\end{matrix}} \rightarrow \boxed{\begin{matrix}\text{dT}_{\text{en}\rightarrow\text{es}}\\ \text{T}\end{matrix}} \rightarrow \boxed{\begin{matrix}\text{dG}_{\text{es}}\\ \text{G}\end{matrix}} \rightarrow \text{es}$$

Si ahora queremos escribir el sistema de traducción inverso, español–inglés, es decir,

$$\text{es} \rightarrow \boxed{\begin{array}{l}\mathrm{dA_{es}}\\ \mathrm{A}\end{array}} \rightarrow \boxed{\begin{array}{l}\mathrm{dT_{es\rightarrow en}}\\ \mathrm{T}\end{array}} \rightarrow \boxed{\begin{array}{l}\mathrm{dG_{en}}\\ \mathrm{G}\end{array}} \rightarrow \text{en}$$

podríamos sacar ventaja del hecho de que hay grandes parecidos entre los datos lingüísticos de este sistema y los del sistema anterior:

- $\mathrm{dA_{es}} \simeq \mathrm{dG_{es}}$: para analizar el español podemos reciclar una buena parte de los datos que se usaban para generarlo en el sistema anterior;
- $\mathrm{dG_{en}} \simeq \mathrm{dA_{en}}$: para generar el inglés podemos reciclar una buena parte de los datos que se usaban para analizarlo en el sistema anterior;
- $\mathrm{dT_{es\rightarrow en}} \simeq \mathrm{dT_{en\rightarrow es}}$: para transferir de español a inglés podemos usar una buena parte de los datos que se usaban para transferir desde el inglés al español en el sistema anterior (por ejemplo, podemos "dar la vuelta" a los diccionarios bilingües y aprovechar muchas entradas).